Apellido y Nombre:	
Carrera:	DNI:
Llenar con letra mavúscu	la de imprenta GRANDE

Universidad Nacional del Litoral Facultad de Ingeniería y Ciencias Hídricas Departamento de Informática Algoritmos y Estructuras de Datos

Algoritmos y Estructuras de Datos. Examen Final. [20 de Diciembre de 2007]

[Ej. 1] [Clases (mínimo 60%)]

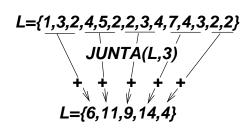
Escribir los métodos que se indican del TAD list<> insert(x,p), erase(p), retrieve(p)/*p, next()/p++, list() y ~list() implementados por celdas enlazadas por punteros o cursores.

[Ej. 2] [Programación (mínimo 40%)]

a) [junta]

Escribir una función void junta(list<int>&L, int n); que dada una lista L, agrupa de a n elementos dejando su suma (ver figura).

Restricciones: No usar ninguna estructura auxiliar. Prestar atención a no usar posiciones inválidas después de una supresión. El algoritmo debe tener un tiempo de ejecución O(n), donde n es el número de elementos en la lista original.



b) [ord-nodo]

Escribir una función predicado bool ordnodo(tree<int> &A); que verifica si cada secuencia de hermanos del subárbol del nodo n (perteneciente al árbol ordenado orientado A) estan ordenadas entre sí, de izquierda a derecha. Por ejemplo, para el árbol (3 5 (6 1 3) (7 4 5)) debería retornar true, mientras que para (3 9 (6 1 3) (7 4 2)) debería retornar false, ya que las secuencias de hermanos (9 6 7) y (4 2) NO están ordenados. Se sugiere el siguiente algoritmo: para un dado nodo retornar false si: 1) sus hijos no estan ordenados o 2) algunos de sus hijos contiene en su subárbol una secuencia de hermanos no ordenada (recursividad).

c) [es-completo]

Escribir una función predicado bool es_completo(btree<int> &A); la cual retorna verdadero si el árbol binario es completo.

[Ej. 3] [operativos (mínimo 70%)]

- a) [rec-arbol] Dibujar el árbol ordenado orientado cuyos nodos, listados en orden previo y posterior son
 - ORD_PRE = $\{C, Z, Q, R, A, M, P, K, L, T\}$,
 - ORD_POST = $\{Z, A, P, M, K, L, R, T, Q, C\}$.
- b) [misc-arbol]: Dado el árbol (c q (t (r u (v z)))),
 - 1) Cuál es el nodo que está a la vez a la izquierda de v y no es descendiente de r?
 - 2) Particione el árbol con respecto al nodo q, es decir indique cuales son sus antecesores y descendientes propios, derecha e izquierda.
- c) [colorear-grafo]

 \mathbf{w} \mathbf{z} \mathbf{r} \mathbf{t} \mathbf{v} \mathbf{z} \mathbf{v} \mathbf{v}

Colorear el siguiente grafo, utilizando una estrategia heurística para tratar de usar el menor número de colores posibles.

Apellido y Nombre:	
Carrera:	DNI:
I lener con letre mayiecu	ile de imprente CRANDE

Universidad Nacional del Litoral Facultad de Ingeniería y Ciencias Hídricas Departamento de Informática Algoritmos y Estructuras de Datos

[Ej. 4] [Preguntas (mínimo 70%)]

a) Sea L una lista conteniendo los elementos (1, 3, 4, 2, 5, 6). Después de aplicar las siguientes líneas

```
list<int>::iterator p,q;
p = L.begin();
q = ++p;
p = L.erase(q);
p++;
q = p;
q++;
```

indique para cada uno de los iterators p, q si es válido o inválido y en el primer caso el valor que almacena (es decir p y q).

- b) Explique que quiere decir la propiedad de "Transitividad" de O().
- c) ¿Cuál es la "Regla del producto" para O()?.
- d) ¿Cuándo se dice que dos tasas de crecimiento del orden del tiempo de ejecución son equivalentes?
- e) ¿Cual es la signatura de la función insert() en listas STL?. Diga que es cada una las variables y cual es el tipo de retorno?. En caso de que haya varias signaturas explique una de ellas.
- f) Explique con un ejemplo la condición de "prefijo" de un código.
- g) ¿Cuál es el costo de inserción en tablas de dispersión abiertas con listas desordenadas en el peor caso?. Diga cual es ese caso.
- h) ¿Que condiciones debe satisfacer un árbol binario para ser "árbol binario de búsqueda"?
- i) ¿Que quiere decir que un algoritmo de ordenamiento sea "estable"?.
- j) ¿Por qué se les dice "lentos" a ciertos algoritmos de ordenamiento?. Enumere los algoritmos lentos que conoce. Enumere los algoritmos rápidos que conoce.